



<b>CUSTOMER REFERENCE</b>	<b>CUSTOMER</b>	<b>INDUSTRY</b>	<b>SOLUTION</b>	<b>1/5</b>
	SOKA-IT	construction industry	natclipse, natcreator, lifecycle-manager	

## Natural upgrade for Bea and Carl

**SOKA-BAU selects the natclipse development environment, lifecyclemanager version and configuration management tool, and template-based generation tool natcreator from innoWake**

The service provider SOKA-IT has been meeting all of the IT needs of SOKA-BAU - the holiday savings and equalization fund for the construction industry and the supplementary pension fund of the building industry - since over 40 years. The SOKA-IT solutions are used by almost 70 000 construction firms with a total of over 620 000 employees and around 420 000 retirees in Germany. The efficient support and continued development of the two Natural-based core applications – premium collection for the insurance funds (Bea) and pension administration (Carl) – play a central role. The natclipse development environment from innoWake, which is based on Eclipse, was introduced in order to provide a future-proof and cost-effective foundation for these tasks. The professional IT company also selected the version and configuration management tool lifecyclemanager as well as the template-based code generation tool natcreator.

### Customer Facts



#### Company

SOKA-IT, a division of SOKA-BAU

#### Headquarters

Wiesbaden

### Savings potential of about two million euros

SOKA-IT has been using the Java, Natural, and ABAP programming languages to develop applications for the insurance industry for years. With a total of around 70 000 source files between them, the »Bea« and »Carl« applications are among the biggest applications in the Natural environment. In addition to continuous support and development, the software engineers also take the topics of future-proofing and efficient IT operation into consideration at all times. For this reason, a large IT project was started in 2005. In order to realize savings of around two million Euros per year and to update the core technology of the applications to the state of the art, the existing mainframe environment running on BS2000 was to be replaced by the UNIX operating system.

»We wanted to take advantage of the fact that UNIX is not only closely tied to the C programming language, but is also an excellent option for Natural applications«, Jürgen Choquet, chief department

manager for IT application management at SOKA-IT, explains the reasoning.

### Every migration project has its perils

Various platform incompatibilities arose during the course of this migration strategy. As an example, the CASE (Computer Aided Software Engineering) tool Predict Case by Software AG, which had been used very intensively for 18 years and was purely host-based, would not run on UNIX. An alternative had to be found. Because at that time, Predict Case was used to manage around 25 000 functional system building blocks and 45 000 text objects, including detailed references to the 13 000 Natural program objects which were compiled from

»It would certainly have been possible to simply port the installed programs. But this would have resulted in the loss of all referencing and the reusability of the Predict Case modules.«

*Wolfram Hassel, IT application management  
group manager at SOKA-IT*

them. In addition, the CASE tool was also used for the transparent administration of all development templates. »It would certainly have been possible to simply port the installed programs. But this would have resulted in the loss of all referencing and the reusability of the Predict Case modules. Considering that a single system module is used in up to 150 programs, for example, the loss of the cross-linking logic and genericity would have been a tragic loss«, Wolfram Hassel, IT application management group manager at SOKA-IT, describes the problems involved in a technology changeover.

So the first step was to start searching for a suitable alternative. Introduction of the SPoD development environment with the individual extension for handling the Predict Case details was discussed. Eclipse-based solution alternatives were also included for evaluation by the project team. After all, Java had been used as a programming language for years. After analyzing a number of tools from several suppliers, the solution from the South-German technology company innoWake was selected at the end of 2007: the eclipse-based development environment natclipse, the version and configuration management tool lifecyclemanager including Subversion, and the template-based code generation tool and eclipse-plugin natcreator.

### Development tool symbiosis

There were several reasons for this decision. Among other things, natcreator is able to reproduce the given Predict Case structures – both with regard to use of the logical functional system building blocks and the display of existing program templates. The Eclipse plugin natclipse, on the other hand, features all the advantages and continuous improvement of the Eclipse IDE. A fully developed code editor offers valuable functions to speed everyday work, such as intelligent code completion, displaying dependencies in the source

code, or displaying the interaction of different modules. A clear overview of the programs and their structures is also available at all times. „This granular referencing function was particularly important in selecting the right tools«, according to Hassel. In addition, natclipse also works well with a version management tool. This allows for efficient development even when larger teams are involved, since, for example, overwriting of other’s source code is automatically prevented. All changes are synchronized and archived in the version management tool. The lifecyclemanager provides the full range of version and configuration management functions. This allows for deployment to the respective environment (development, acceptance testing, and production) including change documentation at

the click of a button. Last but not least, natclipse and natcreator are designed to work together.

»natcreator is like a pre-compiler which the developer doesn’t even notice. You load a program, and the tool automatically takes

action when it has to insert various logical building blocks into the program. Loading, monitoring, and compiling are fully automated«, Wolfram Hassel describes the symbiosis of the two IT systems. In today’s environment, this smooth interaction provides more than extreme ease of development. Great time and cost benefits, coupled with a reduced error rate, also arise through automatic code generation on the basis of standardized modules.

»So we managed to complete a double spin between jumping from the BS2000 world and landing in the Unix environment«

*Jürgen Choquet, chief department manager for IT application management at SOKA-IT*

### **Phase 1 of the migration: A new environment is required!**

Introduction of the new development environment was closely coupled with the platform changeover to the UNIX environment. The key connecting element was the development of a new library architecture. In the BS2000 environment, each application had its own Natural library. This applied both to the development and acceptance testing target environments, and in part to production. This concept was turned on its head for porting to UNIX. The project team transferred all installed program objects to a central core library which mirrored the production environment. Then all Natural objects which were in development or acceptance testing at this time were transferred to the corresponding target branches. The volume to be migrated was reduced by around 50 percent in this manner, since all redundancies in the Natural objects were identified and eliminated. Result of the new concept: Development only contains the new or modified objects and production. During acceptance testing, however, the developers only see those parts of the program which have changed in comparison to production and need to be tested. Whenever a software engineer assigns a new status (development, acceptance testing, production) to an object, that object is transferred to the new

target environment and deleted from the old one. The migration to natclipse was implemented on the basis of this new core library. »So we managed to complete a double spin between jumping from the BS2000 world and landing in the Unix environment«, Jürgen Choquet describes the porting preparation process.

### Phase 2: Populating the new development environment

Special migration programs were developed by innoWake in conjunction with SOKA-IT to perform the actual migration of the new core structure into the natclipse-integrated object management system Subversion. These applied the central library to the installed programs, searched them for logical functional system building blocks, extracted the results, and inserted the corresponding references. The template-based generation tool natcreator was expanded at the same time in order to display and use the logical functional system building blocks from Predict Case. The new environment went live on August 4, 2008 after a total project time of just eight months.

»If one also considers the cost reduction achieved through this changeover, then it is easy to see that this was one of the most important projects for SOKA-IT«

Wolfram Hassel, IT application management  
group manager at SOKA-IT

The two environments ran in parallel until the summer of 2009. During this period, both natclipse and the integrated components lifecyclemanager, Subversion and natcreator served both environments at the same time: from development through acceptance testing to production. »This parallel approach allowed us to eliminate virtually all risk from the changeover. Compiling errors which occurred could be analyzed and eliminated in a targeted manner«, Choquet says.

In the meantime, the project to replace BS2000 was completed successfully, on time and on budget, and with almost no problems. The BEA and CARL applications have been running in a UNIX (Solaris) production environment since June 22, 2009. »The final changeover from BS2000 to UNIX took place over a weekend without any significant constraints on the users. The significant performance improvements expected as a result of testing were confirmed after a short time in production operation and were particularly noticeable in the batch programs. If one also considers the cost reduction achieved through this changeover, then it is easy to see that this was one of the most important projects for SOKA-IT«, Wolfram Hassel added.

### Nobody is missing the old environment

SOKA-IT considers the migration project to be a complete success – and not just in light of the enormous cost savings. Now the service providers can also unify the development paradigms and development processes between Natural and Java. »The efficiency and quality of the entire process was improved. The software engineers are

experiencing this as well. Nobody is mourning the old environment«, Choquet summarized. And the positive impact was not limited to the products of the IT partner. »The innoWake employees were always present and accompanied the project phases with competence and dedication. It was not possible to distinguish them from the in-house personnel«, Jürgen Choquet and Wolfram Hassel agree.

innoWake gmbh  
IT-Tower  
Robert-Bosch-Str. 1  
D-89250 Senden

Tel: +49.7307.92190.0  
Fax: +49.7307.92190.20

The logo for innoWake, featuring the word "innoWake" in a stylized, blue, lowercase font with a wave-like underline.

info@innowake.de  
www.innowake.de

